

XLR8 Your 4P/III

Model 3 Mode, the 4P, and XLR8er

by Gary Lee Phillips

Some folks, myself included, consider Tandy's Model 4P microcomputer to be the most flexible and powerful Z80-based system ever sold to mere mortals. The addition of an XLR8er expansion board from H.I.Tech Inc. can really turn the 4P into a "mean machine" capable of outshooting most MS-DOS systems currently in use.

However, if you use the model 3 mode of your 4P very much, there is a small nuisance about installing XLR8er. The XLR8er replaces your Z80 CPU with the more advanced HD64180. The HD64180 can address more memory and run at higher speeds under software control (it does not require changing the actual clock rate, unlike the Z80 design of the models 4 and 4P). When you reset or first apply power to the XLR8er system, it defaults to its slowest possible speed, which is somewhere near the speed of the old model 1 TRS-80. The boot ROM in a model 4P makes certain assumptions, including a CPU clock rate of 2.0 MHz or thereabouts, which results in a serious problem: if you try to cold boot the XLR8er into model 3 mode, it will usually fail while trying to read the ROM image file from disk. The typical messages will be "Lost Data Error" or "ROM Image not found on Drive 0."

There are various ways to get around this. The one recommended to me by H.I.Tech was to load the ROM image after booting in model 4 mode. You can use the SET180 command to speed the machine up to normal or faster speed, then use the BOOT command of LS-DOS 6.x. Hold down the "P" and "F3" keys, and the machine will load MODEL4/III from your LS-DOS 6.x system disk, and then prompt you for your LDOS or TRSDOS 1.3 master disk followed by enter. This usually works, though sometimes the machine comes up with the "Cass?" prompt instead of reading the model 3 master diskette. (I recover from this situation by pressing enter twice to get to ROM Basic, then using a SYSTEM command, followed by /O and enter. LDOS will then initialize.) Hardly a friendly solution, right? But it is tolerable if you don't use model 3 mode very often. Unfortunately, I support a bulletin board sys-

tem that runs in model 3 mode because it was originally set up on a real model 3. While developing and testing code for this system, I frequently have to reboot after crashing the DOS. Swapping disks and booting LS-DOS, then rebooting and reloading the ROM image each time gets downright irritating.

With a little encouragement from Roy and others on CompuServe, I set out to find the reason for the XLR8er's inability to read MODEL4/III cold. The problem turns out to be merely one of speed: the code in the 4P boot ROM is not fast enough to keep up with a double-density diskette when the CPU is running in its slowest mode. Obviously the answer is to get the HD64180 kicked up to a higher speed before trying to load MODEL4/III. This could be done by modifying the boot ROM (i.e., patch the code and burn it onto a new EPROM, open the 4P and swap chips...not a simple job), or by patching the boot sector of LDOS 5.x to include an initialization routine. The latter seemed reasonable as an approach until I realized that MODEL4/III has to be loaded before the boot sector gets control. It would be necessary to make the boot sector look like a model 4 boot sector, initialize the CPU, and then boot again, this time fooling the ROM into thinking it needed MODEL4/III, even though the boot sector looked like model 4 mode.

Finally it occurred to me that the 4P has the ability to load "customized" ROM images. I haven't seen much written on the subject, but it is documented in the 4P Service Manual from Radio Shack. By default, the system will load the first file it finds in the directory with a name of MODELx/III where the "x" can be any valid character. If you hold down any key from "A" through "G" during the reset, the "x" will be replaced by the specified key, and that specific file will be loaded. Why not write a very short initialization program, and put it on the disk with a name like MODELB/III? Hold down the "B" key when you reset the computer, and let the MODELB/III file load and execute. Since it is very short (only one sector) the odds are much better that it will load without error even with the slowed-down CPU. This program can then speed up the XLR8er and reboot, letting the default MODEL4/III file be loaded instead. Thus, XBOOT was born.

This short program can be assembled by any model 4 or model 3 Z80 assembler. Be sure to generate a /CMD file rather than a /REL file if your assembler has relocating capability.

Copy the output file to your LDOS master diskette (not the original, of course, but a working backup) giving it the name MODELB/III. Make sure MODELA/III is also present on the disk. Now put the INSXBOOT/JCL file on the same LDOS diskette. Get booted up in LDOS, using whatever method you have to in order to get the ROM image loaded this one time. Using the disk with MODELB/III and INSXBOOT/JCL as your system diskette, issue the command: DO INSXBOOT (D=1). When you are prompted, insert a scratch diskette into drive 1, and INSXBOOT will build a new master copy of LDOS, with the MODELx/III and MODELB/III files in the best possible directory slots for successful loading. Since the boot ROM reads the directory of drive 0 sequentially, the farther the MODELx/III file is down the directory list, the better the chances are of getting a lost data error before it can be found or read. INSXBOOT puts a temporary patch on SYS2/SYS to force these files into the top of the directory, rather than letting them fall into random positions. Once you have built a master diskette using INSXBOOT, you can use QFB to copy it as often as you wish and these crucial files will remain in the proper positions. The LDOS or LS-DOS BACKUP utility will also keep the files in place, but only if you are doing mirror image backups. Selective backup or reconstructs will redistribute the files randomly in the directory, so beware.

To boot using the new master diskette, put it into drive 0. Hold down the "B" key on the keyboard, and press the reset button. If you were already running in model 3 mode, you will also need to hold down the "L" key this time to force the ROM image to be reloaded. If you get a "Lost Data Error" message, just try again. It usually works on the first or second try. Once MODELB/III is loaded, it will prompt you to press enter again. At this point, you can use any added key presses desired (see your model 4P manual for details) after pressing enter, and MODELA/III should load successfully and pass control to LDOS.

If instead, you find MODELB/III has loaded a second time, the files are in the directory in the wrong order. MODELA/III must appear first on the directory (you can check this by using DIR :d (O=N) to get the unsorted directory listing). If you ran INSXBOOT, it will have taken care of this for you.

Now what, you may ask, about TRSDOS 1.3? I hope none of our readers are still dependent on this very weak DOS for their model 3 mode operations. However, TRSDOS 1.3 does have one

or two useful features. In particular, I have used it when I had to get files from a model 1 diskette without destroying that diskette (remember, if you use the REPAIR command of LDOS or LS-DOS, the model 1 can't read the diskette any more). You can do a two-step conversion, moving the files first to TRSDOS 1.3 and then to LDOS or LS-DOS, and the original diskette will remain unchanged.

However, TRSDOS 1.3 has some definite problems with the XLR8er. Besides the same inability to cold boot that we just solved for LDOS, TRSDOS 1.3 contains instructions that will not execute correctly on the HD64180 CPU, causing the system to reboot when you try to execute a program, or use the BACKUP command. These instructions are of the type that "split" the index register IX or IY, loading the two halves separately. They are not standard Z80 instructions, but they do work on every Z80 chip I've ever tried them on. Tandy apparently used them in TRSDOS 1.3 as a "security" measure to make it harder to disassemble. I am aware of two places in TRSDOS 1.3 that contain these instructions. The following patch commands will eliminate them (note that this is a TRSDOS 1.3 patch, and NOT an LDOS patch, so please execute it under model 3 TRSDOS!)

```
PATCH *2 (ADD=4EBF,FIND=FD2EFFE1,CHG=FD21FF42)
PATCH *2 (ADD=4EC3,FIND=F1F1FD2642,
          CHG=E1F1F10000)
PATCH *7 (ADD=5284,FIND=DD2662DD2E24,
          CHG=000000000000)
PATCH *7 (ADD=528A,FIND=DD7E00,CHG=003EFF)
```

The patch to SYS7, by the way, also eliminates the limited backup feature of TRSDOS 1.3. Once these patches have been applied, TRSDOS 1.3 appears to execute normally on a 4P equipped with XLR8er. If you find other places where a patch is required, please let us all know!

If you want to boot a TRSDOS 1.3 diskette cold, requiring a load of the ROM image file, you will need to get both MODELA/III and MODELB/III onto the diskette so that they appear at or near the top of the directory, with MODELA before MODELB. Since TRSDOS 1.3 does not allocate directory entries randomly, this is easy to achieve by deleting the top files and then using CONVERT to move the MODELx/III files onto the disk from a 35-track SSSD LDOS diskette. Then copy the other files you deleted back onto your diskette from a backup copy. The TRSDOS 1.3 BACKUP command always produces a mirror image, so once you

get the files into the right position, BACKUP will keep them there.

Notes on the XBOOT program code:

The HD64180 has additional instructions that can be used to address its internal control registers. However, these instructions do not have to be used. The ordinary Z80 instruction, OUT (C),A or IN A,(C) will address the internal registers IF the value in the B register is zero. I have taken advantage of this fact in order to write an initialization routine that can execute harmlessly on a regular Z80, so if you happen to boot your special XLR8er system diskette on a 4P that does not have the HD64180 installed, it will still work. This diskette will also boot normally on a regular model 3 or a model 4, which would not have been the case if I had chosen to patch BOOT/SYS instead of building the special MODELB/III file.

When XBOOT (MODELB/III) gets control, it sets the dynamic memory refresh rate to every 40 t-states rather than the default value of 10. It also sets two wait states on every memory access, and two wait state on every I/O port access. This produces an effective CPU speed slightly faster than the standard model 3. If you wish to experiment with other speeds, you can change the values that are loaded to the RCR and DCNTL ports appropriately. You will need Hitachi publication #U77, HD64180 8-Bit High Integration CMOS Microprocessor User's Manual, in order to select the appropriate values. You can obtain this useful book from H.I.Tech or from Hitachi. H.I.Tech also has a version of SET180 that executes under LDOS. (I wrote my own instead, using the disassembly of the TRSDOS/LS-DOS version as a model.) XBOOT also removes the write protection from the ROM image area of RAM, and pokes a zero value into 3000H. This is one of the addresses checked by the boot ROM to make sure that the ROM image is not already present, and since it is not a C3H as expected, a full reload of MODEL A/III is forced. XBOOT then pages in the boot ROM by loading a 01H value to port 9CH. Since the boot ROM was already executing when XBOOT received control, the necessary RAM areas are still properly initialized, and XBOOT can make use of boot ROM functions, such as the message display function accessed via RST 18H. These functions are documented in the model 4P service manual. After the CPU is initialized and the user gives the go ahead by pressing enter, control is passed to the boot ROM, which then will

attempt a load of the "real" ROM image, MODEL A/III.

Notes on INSXBOOT/JCL

Note that INSXBOOT performs a temporary patch to SYS2/SYS. This patch works on LDOS 5.3 only. If you are installing on an earlier version of LDOS, you will have to make sure you have a correct modification of the patch. Likewise, a similar patch could be made to LS-DOS/TRSDOS if desired, but the patch would not be identical. LDOS 5.3 normally selects a starting place in the directory for creating a new file by looking at the clock heartbeat. The result is random distribution of files across the available directory entries, which is generally desirable. However, in our case we want to get MODEL A/III and MODEL B/III into the very first available directory record. INSXBOOT insures this by forcing the selection to a hard-coded offset for each of these two files, and then restores the original algorithm to SYS2. Please don't run INSXBOOT on your master LDOS diskette! Use a backup copy, just in case the patch causes problems.

Thanks to Roy Soltoff, Joe Kyle Di Pietropaolo, and Adam Rubin for valuable suggestions and bits of information that went into this project. I'd be happy to hear about any improvements, or to try to assist you with any problems you experience with the code. I can be reached on Compuserve's LDOS forum, and my user identification is 72425,354.

```

. INSXBOOT -- this JCL will build an LDOS 5.3 system disk
. in the target drive (default is drive :1) such that
. the MODEL A/III and MODEL B/III files are placed in the
. first sector of the directory. To execute this procedure
. you need a backup copy of your master LDOS 5.3 disk in
. drive 0, with the two ROM image files copied onto it.
. Invoke with DO INSXBOOT (D=drive) where 'drive' is
. the number of the target drive. The optional parameter
. (DS) can be used if the target drive is double-sided.
.   -Gary Phillips, July, 1987.
.
//IF DS
//ASSIGN S=2
//ELSE
//ASSIGN S=1
//END
//IF D
. Target drive is :#D#.
//ELSE
//ASSIGN D=1
. Default drive :1 will be used.
//END
//PAUSE Insert blank disk into drive :#D#, <Enter> when ready.
format :#D# (sides=#S#,name="LDOS-530",mpw="PASSWORD",dden,abs)
backup sys0/sys:0 :#D# (s)
backup sys$/sys:0 :#D# (s,new)
. Now installing temporary patch to SYS2/SYS
patch sys2/sys.system:0 (d03,d4=3e 40 00:f03,d4=3a 88 42)
backup modela/iii:0 :#D# (i)
patch sys2/sys.system:0 (d03,d4=3e 60 00:f03,d4=3e 40 00)
backup modelb/iii:0 :#D#
. Removing temporary patch to SYS2/SYS
patch sys2/sys.system:0 (d03,d4=3a 88 42:f03,d4=3e 60 00)
. Copying remaining files to target disk
backup $:0 :#D# (s,i,new)
. Installation completed. To boot LDOS 5.3 with XLR8er 4P,
. use new system disk in drive :0, and hold down "B" while
. pressing the RESET button. If you get a "Lost Data Error"
. just try again. It usually works within two tries.
. To create additional copies of the new disk, you can use
. QFB to avoid running this process again.
//EXIT

```

MISOSYS MRAS-1.0 07/12/87 13:35:24 XLR8ER LDOS BOOT

Page 00001

```

00002 ;-----
00003 ; XBOOT -- assist program to initialize the
00004 ; HD64180 CPU in an XLR8er board before
00005 ; attempting to load the MODEL A/III file
00006 ; on a 4P. Assemble this code, and rename
00007 ; it to MODEL B/III. Copy it to your boot
00008 ; diskette for LDOS or TRSDOS 1.3. When
00009 ; booting "cold" into model 3 mode, hold
00010 ; down the B key when you press reset. Then
00011 ; when you are prompted to press <Enter>,
00012 ; do so and proceed with normal booting
00013 ; procedure, including any special keypress
00014 ; you may require.
00015 ;

```

```

00016 ;      For best results, the directory entry for
00017 ;      this file should be in the first directory
00018 ;      sector, immediately after that for MOOELA/III.
00019 ;      This can be achieved by zapping OIR/SYS or
00020 ;      by using the INSXBOOT/JCL file.
00021 ;
00022 ;      Gary Lee Phillips, CIS IO [72425,354]
00023 ;      Midsummer, 1987
00024 ;-----
00025 ; Define the ports we need to use:
00026 ;-----
0032 00027 OCNTL EQU 32H      ;dma/wait cntl port
0036 00028 RCR EQU 36H      ;refresh cntl port
0084 00029 OPREG EQU 84H    ;memory mapping cntl
009C 00030 BOOT EQU 9CH     ;boot rom cntl port
00031 ;-----

8000 00033 ORG 8000H      ;above everything
8000 00034 XBOOT EQU $    ;else that we use
8000 F3 00035 DI          ;just to be safe

00037 ; Set up the internal 64180 port values:

8001 0600 00039 LO B,00H      ;simulate in0/out0
8003 0E36 00040 LD C,RCR      ;but z80 compatible
8005 ED78 00041 IN A,(C)      ;get refresh cntl
8007 F602 00042 OR 02H       ;CYC1 = 1
8009 E6FE 00043 ANO 0FEH     ;CYC0 = 0
800B E079 00044 OUT (C),A     ;set 40 state refresh
8000 0E32 00045 LD C,DCNTL    ;now get wait state
800F ED78 00046 IN A,(C)     ;cntl bits and set them
8011 F690 00047 OR 90H       ;MWI1 = 1, IWIO = 1
8013 E69F 00048 AND 9FH      ;MWIO = 0, IW11 = 0
8015 ED79 00049 OUT (C),A     ;2 mem waits, 2 i/o

00051 ; Set up the 4P memory control ports:

8017 3E01 00053 LD A,01H      ;remove memory write
8019 0384 00054 OUT (OPREG),A ;protection
801B AF 00055 XOR A          ;make sure ROM image
801C 320030 00056 LO (3000H),A ;gets (re)loaded
801F 3E01 00057 LD A,01H     ;get boot ROM resident
8021 039C 00058 OUT (BOOT),A ;in low memory

00060 ; Use boot ROM routine to display banner message:

8023 B7 00062 OR A          ;reset Z flag bit
8024 219E80 00063 LO HL,MVECT ;point to msg block
8027 0F 00064 RST 18H       ;display prompt

00066 ; Now wait for user to press <Enter>:
00067 ; (If you prefer not to have to press a key to continue
00068 ; the bootstrap operation, omit the next 3 instructions.

8028 00070 KSCAN EQU $
8028 3A4038 00071 LD A,(3840H) ;read kbd matrix
802B CB47 00072 BIT 0,A       ;enter key?
802D 28F9 00073 JR Z,KSCAN   ;no, so wait

802F C30500 00075 JP 0005H    ;do normal boot

```

```

8032 58      00077 MSG1  DB      'XLR8er Boot Assist '
8045 76      00078      DB      'version 1.0',00H
8051 20      00079 MSG2  DB      ' by Gary L. Phillips,'
8067 20      00080      DB      ' June, 1987.',00H
8074 50      00081 MSG3  DB      'Press <ENTER> to '
8085 63      00082      DB      'continue with bootstrap.',00H

809E 4000    00084 MVECT  DW      1*64,MSG1,-2
80A4 8000    00085      DW      2*64,MSG2,-2
80AA 0001    00086      DW      4*64,MSG3,-1

8000          00088      END      XBOOT

```

```

@@1          0000 @@2          0000 @@3          0000
@@4          0000 BOOT          009C DCNTL          0032
KSCAN        8028 MSG1          8032 MSG2          8051
MSG3         8074 MVECT          809E OPREG          0084
RCR          0036 XBOOT          8000

00000 Total errors
39884 Free space

```

MISOSYS

SOFTWARE TO STRETCH YOUR TRS MODEL 4

PRO-WAM™ Release 2 Window and Application Manager

Our applications turn your 128K Model 4 into a sophisticated business or personal machine rivaling the best of them. Because easily installed PRO-WAM comes with many useful and powerful menu-driven time savers and work organizers. PRO-WAM is accessed with a single keystroke; its EXPORT/IMPORT functions allow you to move data across windows between programs.

- Address CARDS, LABELS, and new HEADINGS for display and export
- Improved BRINGUP tickler file; new PRINTING and sorting
- Improved CALENDAR flags BRINGUP items visually on screen
- Ten 3 x 5 CARD files with FORMS and FIELDS using reverse video
- New virtual PHRASE access for export
- New TODO list manager with "who does it"

PRO-WAM [M-51-025]\$74.95 + \$5 S&H

Upgrade your old PRO-NT/PRO-WAM to release 2; return your master disk with \$24.95 + \$5 S&H



MISOSYS has been supplying the TRS-80 community with professional quality software since 1978; that's over eight years of experience captured in a host of other software products ranging from language compilers and assemblers, line crafted utilities, other applications software, and operating systems. We also publish a magazine, THE MISOSYS QUARTERLY, which is available on a subscription basis. Call or write us for a complete catalog of our product line.

LB —A flexible data manager

Easily used by anyone for managing their data. It's menu driven for ease of use; absolutely no programming needed. Requires a Model 4 with 128K or a hard drive. LB86,™ an MS-DOS version is also available.

- Store up to 65534 records per data base
- Up to 1024 characters per record
- Up to 64 fields per record
- Nine field types for flexibility
- Select and sort on up to 8 fields
- Keep 5 different indexes for data access
- 10 input/update screens per data base
- 10 printout formats per data base
- Extensive on-line help always available

LB [L-50-510]\$74.95 + \$5 S&H



MISOSYS Inc.

P.O. Box 239
Sterling, VA 22170-0239
703-450-4181

VA RESIDENTS ADD SALES TAX
S&H: CANADA ADD \$1; FOREIGN USE S&H '3

ORDER NOW
800-MISOSYS
800-647-6797